

KOREKSI JAWABAN ESAI BERDASARKAN PERSAMAAN MAKNA MENGGUNAKAN FASTTEXT DAN ALGORITMA BACKPROPAGATION

Dian Ahkam Sani¹, M. Zoqi Sarwani²

^{1,2} Teknik Informatika, Universitas Merdeka Pasuruan, Indonesia

email: dian.ahkam@unmerpas.ac.id¹, zoqi.sarwani@unmerpas.ac.id²

Abstrak

Word Embedding adalah bagian dari NLP (Natural Language Processing) yang belakangan ini menjadi topik menarik untuk diteliti. *Facebook AI research* mengusulkan FastText sebagai salah satu metode *word embedding* yang memiliki keunggulan menemukan kata yang jarang ditemui atau OOV (*Out of Vocab*). Algoritma Backpropagation sebagai salah satu metode yang tidak hanya bisa mengklasifikasi namun juga memprediksi dengan arsitektur multilayer yang sering digunakan dengan mencari bobot optimal pada jaringan saraf tiruan. Pada tingkat akademisi penggunaan word embedding sudah banyak dilakukan untuk mencari kesamaan kata ataupun makna. Penelitian ini bertujuan untuk mengoreksi jawaban esai berdasarkan kesamaan makna sebagai alternatif bagi guru dalam menilai ujian. Data yang digunakan adalah data jawaban yang diambil dari pelajaran bahasa Indonesia sebanyak 10 soal dari total 50 siswa. Data jawaban tersebut nantinya akan ditraining dan disimpan kedalam bentuk vektor dengan menggunakan FastText. Sedangkan pemodelan data train sebelumnya menggunakan model corpus wikipedia dengan besaran dimensi vektor 200, *n-window* 5, dan *min-count* 3. Dari proses tersebut maka hasil penelitian menunjukkan bahwa dari 500 data yang digunakan didapatkan kombinasi terbaik dimana data *training* 75% sebesar 375 data dan jumlah data *testing* 25% sebesar 125 data dengan membandingkan nilai Y_{aktual} dengan Y_{prediksi} menggunakan MAPE (*Mean Absolute Percentage Error*) sehingga nilai MAPE terbaik yang didapatkan pada hasil pengujian proses *training* dan *testing* masing-masing 5% dan 8%. Dari hasil tersebut dapat dilihat bahwa model yang dihasilkan menggunakan algoritma backpropagation untuk melakukan penilaian/koreksi jawaban secara otomatis sangat baik.

Kata kunci: Penilaian esai otomatis, Word Embedding, FastText, Backpropagation.

Abstract

Word embedding is part of NLP (Natural Language Processing), which has become an interesting research topic these days. Facebook's AI research suggests FastText as a way to embed words. This has the advantage of finding OOVs (Out of Vocab). Backpropagation algorithms are widely used in artificial neural networks to find optimal weights as a way to not only classify but also predict using a multi-tiered architecture. Word embedding usage levels have been performed to find similarities in words and meanings. This study aims to modify essay answers based on semantic similarity as an alternative to teachers when scoring exams. The data used are answers to up to 10 questions from a total of 50 students from Indonesian lessons. The response data will later be trained in FastText and saved in vector format. Previous train data modeling used the Corpus Wikipedia model with vector dimensions of 200, n-window 5, and min-count 3. From this process, the results showed that of the 500 data used, the best combination was obtained when 75% of the training data was 375. A set of 25% of test data is 125 data, with the highest MAPE values in Y_{actual} and $Y_{\text{prediction}}$ (mean absolute percentage error) obtained from the test results of the training process and the test process being 5% and 8%, respectively. From these results, we can see that the model generated using the backpropagation algorithm is very suitable for automatic scoring / correction of responses.

Keywords : Automatic Essay Score, Word Embedding, FastText, Backpropagation.

Diterima Redaksi: 30-06-2022 | Selesai Revisi: 20-07-2022 | Diterbitkan Online: 31-07-2022
DOI: <https://doi.org/10.23887/janapati.v11i2.49192>

PENDAHULUAN

Salah satu tugas tenaga pendidik dalam hal ini guru adalah menilai pengetahuan yang

didapat siswa selama proses pembelajaran. Umumnya guru akan menggunakan berbagai instrumen penilaian untuk menilai dan

mengevaluasi siswa. Seperti kuis, ujian dan tugas rumah. Seiring berkembangnya teknologi dan informasi terhadap dunia pendidikan, pembelajaran-pun bisa dilakukan secara daring atau jarak jauh. Melihat nilai peserta didik secara *online*, mengecek keuangan, melihat jadwal perkuliahan, mengirimkan berkas tugas dan sebagainya [1].

Didalam dunia pendidikan biasanya tugas dibagi menjadi dua yaitu tugas pilihan ganda atau tugas esai. Tugas esai merupakan penilaian, pandangan, pendirian, atau evaluasi penulis terhadap suatu hal untuk diambil kesimpulannya biasanya esai ditulis dengan gaya dan ciri personal atau individual penulisnya [2]. Namun, untuk menilai dan menganalisa jawaban-jawaban siswa mungkin agak sulit dan memakan waktu yang lama. Untuk mengatasi masalah dalam penilaian esai maka diperlukan sebuah sistem yang dapat mengoreksi jawaban tersebut dengan waktu yang singkat dan akurat.

NLP (*Natural Language Processing*) merupakan suatu pengembangan teknik komputasi bahasa alami dalam menganalisis dan merepresentasikan teks ataupun lisan untuk mencapai bahasa seperti bahasa manusia. Keterkaitan dan kesamaan semantik berkaitan dengan bidang linguistik khususnya pada bidang NLP yang menarik untuk diteliti. Keterkaitan kata maupun teks memiliki peran penting dalam beberapa task dari NLP dan beberapa bidang terkait seperti *topic modeling*, *sentence classification*, *sentiment analysis*, *text summarization*, *similarity text* dan lain sebagainya [3]. *Similarity text* belakangan ini menjadi topik yang menarik dan banyak diteliti. *Word Embedding* sangat populer dalam bidang NLP [4][5] karena dapat digunakan untuk menggambarkan kedekatan sebuah kata atau sebuah dokumen namun kedekatan kontekstual sesuai dengan data latih yang digunakan dalam pembedukannya sehingga seringkali kedekatan tersebut bukan merupakan makna sebuah kata, melainkan bahwa setiap metode dalam *Word Embedding* dapat digunakan dalam perhitungan kesamaan semantik sebuah kata dari *query* yang diminta oleh pengguna [6].

Pada tahun 2016, *Facebook AI Research* mengusulkan *FastText* sebagai kategori teks. *FastText* memprediksi tag (jenis teks, yang ditentukan oleh anotasi manual) melalui konteks salah satu keunggulannya yaitu dapat menemukan kata yang jarang ditemui karena *FastText* memanfaatkan *sub-word* dari setiap kata [7][8]. *FastText* adalah library yang bisa membantu untuk menghasilkan representasi kata secara efisien dan memberi bantuan untuk meng-kalsifikasikan teks di luar Pustaka. [3][7]

mengklaim bahwa penggunaan *FastText* lebih unggul dalam mengetahui kata-kata di luar Pustaka dan dapat menangani penggunaan Bahasa yang berbeda yang mungkin dalam sebuah corpus tidak tersedia. Algoritma *Backpropagation* adalah salah satu model jaringan saraf tiruan yang dapat digunakan untuk menyelesaikan masalah klasifikasi ataupun prediksi dengan arsitektur multilayer yang sering digunakan dengan mencari bobot optimal pada jaringan saraf tiruan.

Berdasarkan latar belakang diatas yang telah dijelaskan maka pada penelitian ini menggunakan *FastText* dan algoritma *backpropagation* untuk mengoreksi jawaban esai berdasarkan makna. Adapun penelitian yang berhubungan dengan penelitian ini antara lain:

Penelitian yang berkaitan dengan *Word Embedding* telah dilakukan oleh [6] dalam pembuatan model bahasa Indonesia berbentuk vector dengan metode *Word2Vec*. Data yang digunakan adalah data yang didapat dari hasil *crawling* di beberapa portal berita. Hasilnya adalah tercapainya pemetaan vektor bahasa Indonesia berdasarkan kata yang digunakan. Penelitian selanjutnya [9] menggunakan *Word2Vec* untuk menganalisis penyelesaian soal analogi pada bahasa Indonesia. Penelitian ini membandingkan model *Skip-Gram* dan *CBOV* dengan menggunakan data latih *corpus* Wikipedia Indonesia dimensi 100 dan window size 10 serta 12. Hasil yang diperoleh adalah 34% untuk model *Skip-Gram* dan 33% untuk *CBOV*. Penelitian yang dilakukan [10] tentang mengetahui karakter seseorang pada cuitan status di social media dengan menggunakan algoritma PNN. Status diambil dari 50 pengguna facebook dan twitter berupa teks sehingga hasil akurasi yang didapat sebesar 86.99% pada saat training dengan data latih 30 dan 83.66% pada saat testing dengan 20 data sebagai testing. Penelitian berikutnya [11] tentang penilaian jawaban otomatis dengan memanfaatkan kombinasi antara *term frequency* dan *n-gram*. Perbandingan dua metode *cosine similarity* dan *jaccard similarity* dengan kombinasi unigram, bigram, unigram dan bigram. Hasilnya metode *cosine similarity* menunjukkan unigram, bigram, dan unigram bigram pada nilai MAE (Mean absolute Error) 4.23 dan 12.8 sedangkan metode *jaccard similarity* 5.26, 4.70, dan 4.63. Penelitian yang berkaitan dengan koreksi esai otomatis juga dilakukan oleh [12] menggunakan metode SVM (Support Vector Machine) dan LSA (Latent Semantic Analysis) dengan jumlah data 148 yang terdiri dari 4 soal dan 37 jawaban siswa. Pada penelitian ini peneliti menggunakan varian kernel diantaranya RBF, linear, dan

Tabel 1. Dataset penelitian

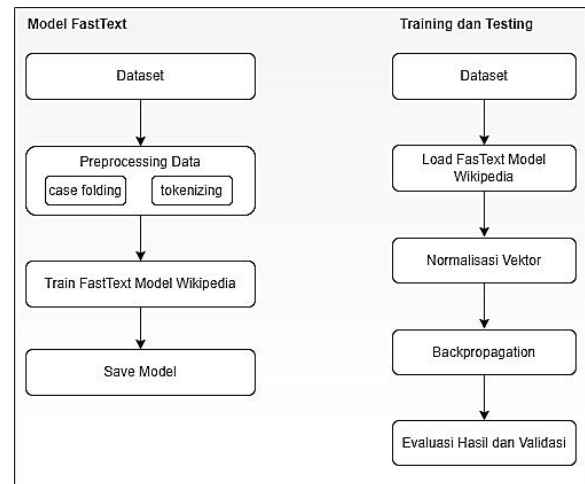
No	Soal	Jawaban	Hasil
1	1	Teks ulasan adalah teks yang berisi pembahasan suatu karya yang	100
2	1	Struktur teks ulasan merupakan susunan yang membangun sebuah teks ulasan sehingga menjadi suatu teks yang utuh.	80
3	1	Teks yg berisi pembahasan suatu karya yg di tunjukkan untuk pembaca mengenai kualitas	80
4	1	Identitas karya yang minimal terdiri dari judul, pengarang, penerbit, tahun terbit, tebal halaman, dan ukuran buku.	80
5	1	Teks ulasan merupakan teks yang berisi pengulasan berita	60
...
50	10	Sepertinya ada 2	10

polinomial, bersama dengan 0, 0.01, 0,1, dan 1 sebagai penalti C. Akurasi yang didapatkan sejumlah 97,297% dengan parameter terbaik C1, kernel RBF dengan parameter C1. Proses pencarian kemiripan kata menggunakan LSA dengan rata-rata persamaan antara jawaban siswa dan guru 72.01%. Penelitian selanjutnya dilakukan oleh [13], peneliti menggunakan metode LSA dan *cosine similarity* sebagai penilaian esai otomatis bahasa Indonesia. Terdapat beberapa tahapan dalam *preprocessing* diantaranya adalah pembersihan data, *case folding*, *tokenization*, *stopword*, *convert negation*, dan *stemming*. Akurasi yang didapatkan dengan metode tersebut dibandingkan dengan penilaian manual adalah 83.3%. Selanjutnya [14] tentang improvisasi *automatic essay scoring* bahasa Indonesia menggunakan *simpler model* dan *richer feature*. Dimana pada penelitian ini menggunakan *pretrained model* BERT *sentence embedding* yang ukurannya lebih kecil dari FastText model. Hasil F1-score yang didapat adalah 82%. Sedangkan Penelitian yang berkaitan dengan performa backpropagation terhadap algoritma supervised lainnya dilakukan oleh [15] dimana algoritma yang digunakan sebagai perbandingan adalah K-NN, SVM, Decision Tree dan Backpropagation dengan jumlah data 14.170. performa akurasi yang diberikan oleh algoritma K-NN adalah 88%, SVM dan Decision Tree memperoleh tingkat akurasi 90%, sedangkan algoritma backpropagation menghasilkan akurasi 100%.

METODE

Pada penelitian ini memiliki beberapa tahapan yang akan dilakukan diantaranya yaitu diawali dengan melakukan pengumpulan data, melakukan *preprocessing* data, FastText *Modelling*, normalisasi vektor dan proses prediksi data dengan menggunakan algoritma

Backpropagation. Gambar 1 menunjukkan alur dari koreksi esai otomatis.



Gambar 1. Alur Penelitian

A. Pengumpulan Data

Tahapan awal yang dilakukan untuk melakukan penelitian ini adalah mendapatkan data jawaban dari 50 siswa/i pada soal esai mata pelajaran bahasa Indonesia kelas 8 sebanyak 10 soal di MTsN Ma'arif Keraton, Kab. Pasuruan. Tabel 1 menunjukkan data jawaban beserta nilai yang diberikan pengajaran di setiap soal.

B. Preprocessing Data

Data yang sudah didapat adalah data asli dimana data masih mengandung banyak variasi dari besar kecilnya huruf, tanda baca, atau kata yang disingkat sehingga perlu dilakukan pengolahan data agar menjadi data yang siap untuk diproses. Pada penelitian ini terdapat beberapa tahapan yang digunakan untuk *preprocessing* data diantaranya yaitu *case folding*, *tokenizing* dan juga pemodelan data corpus wikipedia.

Tahap *case folding* pada penelitian ini adalah untuk mengubah seluruh huruf kapital

menjadi huruf kecil sebagaimana yang ditunjukkan pada tabel 2.

Tabel 2. *Case Folding*

Kalimat	Struktur teks ulasan merupakan susunan yang membangun sebuah teks ulasan.
Case Folding	struktur teks ulasan merupakan susunan yang membangun sebuah teks ulasan.

Langkah selanjutnya adalah *tokenizing*, *tokenizing* adalah tahap pemotongan *string* berdasarkan kata-kata yang menyusunnya atau pemecahan kalimat menjadi kata. Umumnya proses pemecahan kalimat dilakukan pada kalimat yang memiliki spasi dan membuang karakter tanda baca yang nantinya setiap kata yang sudah di proses akan menjadi token. Dari kata yang sudah ditoken selanjutnya akan dibuat model dengan metode FastText. Tabel 3 menjelaskan hasil dari *tokenize* pada kata "struktur teks ulasan merupakan susunan yang membangun sebuah teks ulasan", yang sudah di lakukan proses *case folding* sebelumnya. Setelah proses *tokenize* dilakukan, langkah selanjutnya adalah menambahkan data ke dalam data corpus Wikipedia. Proses ini dilakukan untuk menambah kata-kata baru apabila tidak terdapat di dalam *corpus*. Proses normalisasi kata berakhir sampai proses *tokenizing*. Penelitian ini tidak menggunakan proses *filtering*, *stopword*, maupun *stemming* karena setiap kata yang termasuk proses tersebut bisa menjadi makna tersendiri dan berefek kepada jarak vektor sebuah kata atau makna. Setelah proses *tokenize* dilakukan maka langkah selanjutnya *training* data kedalam data corpus Wikipedia. Proses ini dilakukan untuk menambahkan kata baru apabila tidak terdapat di dalam corpus.

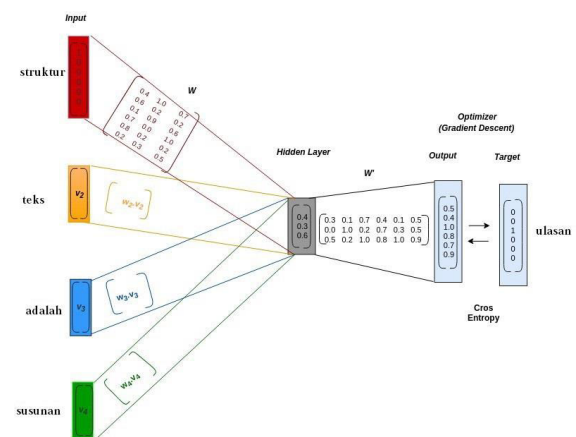
Tabel 3. *Tokenizing*

Kalimat	struktur teks ulasan merupakan susunan yang membangun sebuah teks ulasan.
Tokenizing	"struktur", "teks", "ulasan", "merupakan", "susunan", "yang", "membangun", "sebuah", "teks",

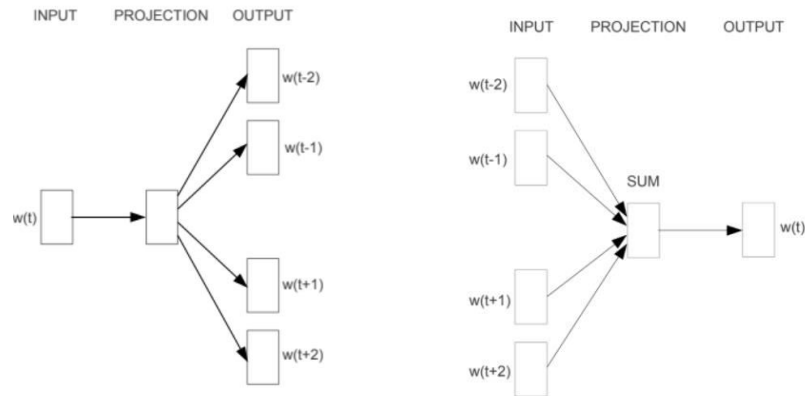
"ulasan", "."

C. FastText Modelling

Penggunaan FastText pada penelitian ini dilakukan untuk mengubah kata menjadi sebuah vektor. Keunggulan FastText adalah bisa menangani kata yang tidak pernah dijumpai sebelumnya OOV (*out of vocab*) karena prosesnya adalah memanfaatkan *sub-word* dari setiap kata [7]. FastText memiliki 2 model arsitektur yaitu CBOW (*Continuous Bag of Word*) dan *Skip-Gram*. Gambar 2 menunjukkan model yang digunakan pada FastText. Model yang digunakan pada penelitian ini adalah CBOW (*Continuous Bag of Word*). CBOW adalah model yang memprediksi kata/output ketika diberikan konteks/input disekitar kata tersebut. CBOW juga mempelajari distribusi kemungkinan kata dari konteks dengan *window* yang sudah ditentukan. CBOW menggunakan konteks sebagaimana tabel 4 sehingga menggunakan *one-hot encoded vector* yang serupa. Tabel 4 menunjukkan contoh penggunaan CBOW dengan konteks/input "struktur teks ulasan adalah susunan teks" dengan nilai *window* = 2. Ilustrasi *forward-backward CBOW* dengan nilai random sebagai inputan sebagai inputan *weight/ bobot* (W dan W'), dengan $w(t-2) =$ struktur, $w(t-1) =$ teks, $w(t+1) =$ adalah, $w(t+2) =$ susunan. Dan $w(t) =$ ulasan sebagai target. Gambar 3 adalah ilustrasi *forward backward training CBOW*.



Gambar 3. Ilustrasi *forward backward training CBOW*



Gambar 2. Arsitektur SkipGram dan CBOW fasttext.

Tabel 4. CBOW dengan 2 window.

Konteks Input, Target	struktur teks ulasan	teks, struktur	ulasan, struktur	
Konteks Input, Target	struktur teks ulasan adalah	struktur, teks	ulasan, teks	adalah, teks
Konteks Input, Target	struktur teks ulasan adalah susunan	struktur, ulasan	teks, ulasan	adalah, ulasan susunan, ulasan
Konteks Input, Target	teks ulasan adalah susunan teks	teks, adalah	ulasan, adalah	susunan, adalah teks, adalah

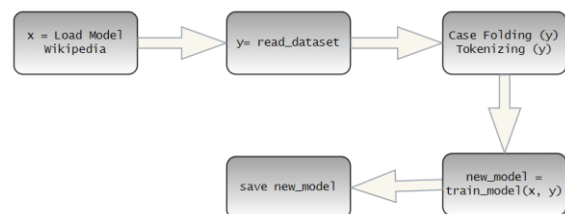
Setelah output diperoleh, selanjutnya adalah melakukan perhitungan nilai error dengan metode *cross entropy* (target-output). Tahap backpropagation dilakukan setelah melakukan perhitungan *error* yaitu dengan menghitung gradient descent terhadap semua parameter yang ada dengan mencari nilai *derivative partial*. Pada tahap backpropagation proses *update* parameter akan berlangsung dengan cara mengurangi atau menambahkan bobot lama dengan nilai *gradient* yang didapatkan hingga tercapai minimum *error* pada *cross entropy*.

Model yang digunakan sebagai *pretrain* pada penelitian ini adalah model yang didapat dari Wikipedia dengan jumlah vektor 300 dengan menggunakan parameter 5 *window* dimana proses perhitungan dilihat dari 5 kata tetangga baik sebelumnya ataupun setelahnya. Tabel 5 menggambarkan contoh nilai vektor pada sebuah kata dengan model wikipedia.

Tabel 5. Vektor kata pada model Wikipedia

get_word_ve	Word	Vektor
-------------	------	--------

ctor	orientasi	0.79
orientasi	orientasinya	0.76
	diorientasikan	0.72
	disorientasi	0.71



Gambar 4. Penambahan vektor kata pada model Wikipedia

Untuk meningkatkan nilai vektor pada model yang sudah dibuat maka perlu membuat model tambahan yang mencakup dataset yang dimiliki. Proses yang dilakukan adalah membaca dataset yang akan dipelajari dengan menormalisasi dataset (*case folding, tokenizing*). FastText tidak menyarankan

stemming karena FastText mempunyai fitur yang dapat membuat sebuah vektor baru dari sebuah kata-kata yang tidak ada atau tidak pernah dipelajari dari *sub-word*, begitupun dengan proses *stopword* karena semakin banyak kata-kata yang dipelajari oleh FastText maka semakin baik modelnya. Gambar 4 menampilkan proses pembuatan model tambahan pada FastText.

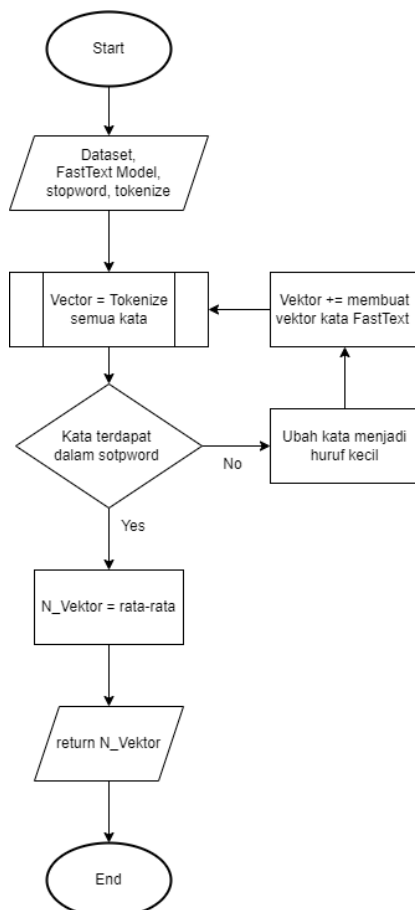
Dari hasil penambahan kata pada model Wikipedia, maka didapatkan hasil sebagaimana table 6.

Tabel 6. Hasil penambahan kata pada model

	Word	Vektor
get_word_ve	orientasi	0.81
ctor	orientasinya	0.79
orientasi	diorientasikan	0.73
	disorientasi	0.72

D. Normalisasi Vektor

Proses pelatihan hingga prediksi menggunakan metode backpropagation akan dimulai dari membaca data yang digunakan, membaca model FastText yang akan mengubah data semula berbentuk *string* menjadi sebuah vector.



Gambar 5. Proses normalisasi vektor

Proses mengubah data menjadi vector diinisialisasi sebagaimana gambar 5.

Proses yang dilakukan dalam normalisasi vector adalah memisahkan (*tokenize*) setiap kata, lalu memproses pengulangan dari setiap kata yang ada. Jika kata yang diproses termasuk di dalam daftar *list* dari *stopword* maka proses dilewati, sebaliknya jika kata yang diproses tidak ada dalam daftar *list* maka kata akan diubah menjadi huruf kecil, lalu dilakukan proses pengubahan kata tersebut menjadi vektor dengan FastText. Setelah semua kata diubah menjadi vektor, maka vektor dari sebuah teks akan dinormalisasi dengan cara vektor dari teks dijumlahkan lalu akan dihitung rata-ratanya [4]. Proses ini akan menghasilkan vektor dari data yang akan digunakan untuk diproses dengan metode backpropagation.

E. Backpropagation

Algoritma backpropagation merupakan salah satu algoritma pada kelompok jaringan saraf tiruan. Algoritma ini bekerja dengan 2 tahap yaitu *training* dan *testing*. Tahap training dilakukan untuk mendapatkan bobot terbaik dengan cara melakukan update bobot pada setiap iterasi sampai mendapatkan nilai error yang kecil, sedangkan tahapan testing dilakukan untuk menguji bobot terbaik yang didapatkan pada proses training. Arsitektur jaringan backpropagation terdiri dari input layer, hidden layer, dan output layer sebagaimana gambar 6. Dimana setiap lingkaran pada setiap layer adalah neuron, setiap neuron memiliki panah masuk dan keluar tergantung pada jenis layernya. Setiap panah mewakili bobot sesuai dengan pentingnya koneksi itu. Layer paling bawah adalah input layer, layer di tengah adalah hidden layer dan yang paling atas adalah output layer.

Pada penelitian ini jumlah input layer sesuai dengan jumlah vektor model yang dihasilkan oleh FastText. Sedangkan jumlah hidden layer sebanyak 3 (128, 32, 8) dan jumlah neuron output adalah 1. Terdapat fungsi aktivasi yang merupakan suatu fungsi untuk mentransformasikan inputan menjadi output tertentu. Selain fungsi aktivasi terdapat juga optimasi *solver* yaitu *Stochastic Gradient Descent* (SGD) yang digunakan untuk mengupdate parameter bobot dan bias. Algoritmanya hanya mengurangi inisial bobot dengan Sebagian dari nilai *gradient* yang sudah diperoleh. Biasanya SGD diwakili oleh *hyperparameter* Bernama *learning rate* (Alpha).

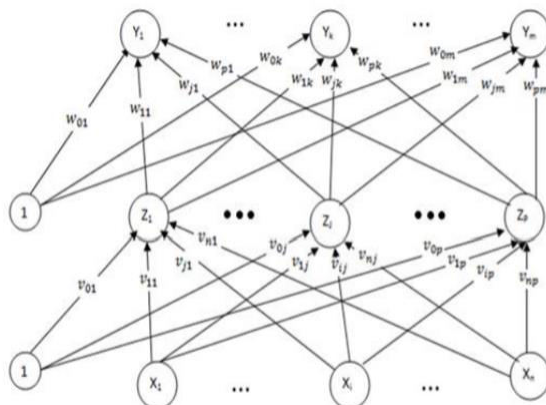
Adapun tahapan perhitungan yang digunakan pada algoritma backpropagation adalah

1. Menentukan parameter learning rate dan epsilon.
2. Lakukan proses pembangkitan bobot secara acak.
3. Lakukan perhitungan pada setiap neuron hidden layer serta membangkitkan fungsi aktivasi ReLU (*Rectified Linear Unit*). Aktifasi ReLU adalah lapisan aktifasi pada model yang menerapkan fungsi $(x) = \max(0, x)$ dimana ReLU intinya membuat batas pada bilangan 0, yaitu apabila $x \leq 0$ maka $x = 0$ dan apabila $x > 0$ maka $x = x$. Turunan fungsi aktifasi ReLU:

$$\frac{\partial y}{\partial x} = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (1)$$

4. Lakukan proses perhitungan pada output layer.

Untuk proses training pada tahap ini lakukan proses update bobot berdasarkan nilai eror. Sedangkan pada proses testing lakukan perbandingan antara output dan target.



Gambar 6. Arsitektur Backpropagation.

F. Evaluasi

Ketepatan prediksi merupakan hal penting untuk dilakukan yaitu bagaimana mengukur kesesuaian antara data yang sudah ada dengan data yang telah diprediksi. Evaluasi merupakan tahapan yang digunakan untuk mengetahui seberapa baik sebuah model. Pada penelitian ini metode evaluasi yang digunakan yakni *mean absolute percentage error* (MAPE). MAPE merupakan sebuah metode untuk mencari persentase *error* dimana Y adalah data aktual sedangkan Y' adalah data prediksi menggunakan rumus:

$$MAPE = \frac{Y - Y'}{Y} \quad (2)$$

HASIL DAN PEMBAHASAN

Setelah dilakukan penelitian dan pengumpulan data, maka pada bab ini dilakukan pengolahan dan analisa terhadap data tersebut. Pengolahan dan analisa dilakukan dengan proses pemilihan data hasil jawaban siswa, preproesing, text modeling dengan fasttext, proses pelatihan backpropagation, analisis hasil pengujian, dan nilai kinerja. Pengolahan dan analisa dijabarkan sebagai berikut:

A. Pemilihan data

Data yang digunakan pada penelitian ini merupakan data hasil jawaban siswa pada soal esay yang telah diberi skor. Jumlah siswa yang mengikuti ujian sebanyak 50 siswa, setiap siswa melakukan pengisian jawaban pada 10 soal sehingga total data yang digunakan sebanyak 500 data. Kemudian data semua jawaban tersebut akan dilakukan proses preproesing yakni mengubah semua jawaban siswa menjadi huruf kecil dan menghapus tanda hubung pada setiap jawaban. Dari data tersebut akan ditentukan mana yang akan dijadikan sebuah fitur dan target. Data hasil jawaban siswa dijadikan sebuah fitur dan skor hasil jawaban siswa akan dijadikan sebuah target.

B. Normalisasi Data

Data yang akan dimasukkan ke dalam metode backpropagation perlu dilakukan text modelling dengan fasttext. Proses ini dilakukan untuk mengubah sebuah kata menjadi vector karena backpropagation tidak memproses data yang berupa text.

Proses pelatihan FastText dilakukan dengan menggabungkan data corpus dari Wikipedia dan hasil jawaban siswa. Penggabungan data dilakukan untuk mendapatkan nilai vector kata yang lebih baik dan memperbanyak kamus kata.

Data hasil jawaban siswa merupakan data yang berupa kalimat. Sedangkan metode fasttext hanya bisa merubah kata menjadi vector, sehingga diperlukan proses normalisasi vector. Proses tersebut dilakukan dengan cara mengambil rata-rata vector dari kumpulan kata pada kalimat [4]. Data vector hasil normalisasi tersebut selanjutnya akan digunakan sebagai input pada proses algoritma backpropagation.

Pada tabel 7 dan 8 ditunjukkan proses normalisasi vector.

Tabel 7. Normalisasi Vektor

Kata	Vektor
Struktur	0.15, 2.1, 2.4, ..., 2,2
Teks	1.29, 1.03, -4.04, ..., 5.32
Ulasan	1.65, 7.39, -1.40, ..., 0.10
Merupakan	-4.89, 7.02, -2.48, ..., 1.56
Susunan	0.41, 0.22, 0.07, ..., -0.06

300 vektor dari setiap kata dijumlahkan lalu dirata-rata agar terbentuk vektor kalimat. Tabel 8 menunjukkan hasil pembentukan sentence vektor.

Tabel 8. Hasil Normalisasi Vektor

Teks	Struktur	teks	ulasan
Merupakan susunan	0.017,	0.027,	-0.012, ..., 0.031

C. Inisialisasi Jaringan

Untuk memulai proses penghitungan backpropagation perlu menentukan arsitektur jaringan. Pada penelitian ini arsitektur jaringan yang dibentuk memiliki 5 lapisan layer yang terdiri dari *input layer*, 3 *hidden layer*, dan *output layer*. Untuk *input layer* jumlah *neuron* ditentukan berdasarkan jumlah vektor yang dihasilkan pada proses *text modelling*, *hidden layer 1* sejumlah 128, *hidden layer 2* sejumlah 32, dan *hidden layer 3* sejumlah 8. Sedangkan pada *output layer* hanya menggunakan 1 *neuron*. Jumlah *neuron output layer* ini didapat berdasarkan target yang ingin dihasilkan oleh backpropagation dimana pada penelitian ini backpropagation digunakan untuk memprediksi skor jawaban siswa. Pada tabel 9 ditunjukkan secara detail arsitektur jaringan dan fungsi aktivasi yang digunakan pada setiap layer pada penelitian ini.

Tabel 9. Arsitektur Jaringan

Nama Layer	Jumlah Neutron	Fungsi Aktivasi
Input	300	-
Hidden 1	128	ReLU
Hidden 2	32	ReLU
Hidden 3	8	ReLU
Output	1	-

D. Inisialisasi Parameter

Inisialisasi parameter merupakan sebuah proses yang dilakukan setelah membentuk arsitektur jaringan. Algoritma

backpropagation memiliki beberapa parameter yang terdiri dari learning read, epoch, dan epsilon. Penentuan parameter pada algoritma backpropagation perlu dilakukan secara tepat karena dapat mempengaruhi model yang akan dihasilkan pada tabel 10 menunjukkan beberapa penentuan parameter yang dilakukan pada penelitian ini.

Tabel 10. Inisialisasi Parameter

Parameter	Nilai
Learning Rate	0,01
Epoch	1000
Treshold	0,001

E. Inisialisasi Bobot

Bobot adalah nilai matematis dari sebuah koneksi antar neuron. Bobot merupakan penghubung antar lapisan yang terdapat pada algoritma backpropagation dimana bobot lapisan masukan ke lapisan tersembunyi pertama, bobot lapisan tersembunyi pertama ke lapisan tersembunyi kedua, bobot lapisan tersembunyi kedua ke lapisan tersembunyi ketiga, dan lapisan tersembunyi ketiga ke bobot lapisan keluaran. Demikian juga dengan bias mempunyai koneksi antar neuron yang sama seperti bobot. Bias merupakan sebuah unit masukan yang nilainya selalu satu.

Setiap kali membentuk jaringan backpropagation, kita perlu menentukan nilai bobot dan bias awal dengan bilangan acak kecil. nilai acak yang dibangkitkan pada penelitian ini antara interval -1 sampai 1. Bobot dan bias ini akan berubah setiap kali kita membentuk jaringan.

F. Pengujian Backpropagation

Proses pengujian backpropagation dilakukan untuk mendapatkan model terbaik. Algoritma backpropagation memiliki dua tahap yaitu tahap training dan tahap testing. Tahap training digunakan untuk mendapatkan model terbaik dan tahap testing digunakan untuk menguji model yang dihasilkan pada tahap training. Pada penelitian ini jumlah data yang digunakan pada proses training dan testing dilakukan dengan cara melakukan pengujian kombinasi terbaik antara data training dan testing. Hasil pengujian data training dan testing ditunjukkan pada tabel 11.

Tabel 11. Kombinasi pengujian data train dan testing

No.	Jumlah data (%)		Nilai MAPE (%)	
	Traini ng	Testi ng	Traini ng	Testi ng
1.	25	75	0	80
2.	40	60	3	25
3.	50	50	3	15
4.	60	40	8	6
5.	75	25	5	8

Pada tabel 11 menunjukkan bahwa kombinasi terbaik untuk data training dan testing masing-masing sebesar 75% jumlah data training dan 25% jumlah data testing. Nilai mean absolute percentage error (MAPE) yang didapatkan pada pengujian kombinasi tersebut sebesar 5% untuk proses training dan 8% untuk proses testing.

G. Analisa Hasil Pengujian

Tujuan utama penggunaan backpropagation adalah mendapatkan keseimbangan antara pengenalan pola pelatihan secara benar dan respon yang baik untuk pola lain yang sejenis (disebut data testing). Jaringan dapat dilatih terus menerus hingga semua pola pelatihan dikenali dengan benar. Umumnya data dibagi menjadi 2 bagian saling asing, yaitu pola data yang dipakai sebagai data training dan data yang dipakai untuk testing. Jumlah data yang digunakan pada penelitian ini sebesar 500 data sehingga berdasarkan hasil pengujian kombinasi jumlah data training dan data testing terbaik yang terdapat pada tabel 6 maka jumlah data training sebesar 375 data dan jumlah data testing sebesar 125 data. Kombinasi jumlah data tersebut merupakan kombinasi terbaik karena dapat menghasilkan nilai kombinasi MAPE terkecil dibandingkan dengan nilai kombinasi lainnya. Dari hasil pengujian kombinasi jumlah data training dan testing dapat kita lihat bahwa ketika jumlah data training lebih kecil dari pada jumlah data testing maka akan terjadi *overfitting*. Hal tersebut terjadi dikarenakan data yang digunakan pada proses training tidak mampu menghasilkan model yang menggambarkan pola keseluruhan data.

KESIMPULAN

Setelah dilakukan analisa dan pengolahan data, maka pada bab ini diambil kesimpulan dan diberikan saran untuk penelitian selanjutnya.

A. Kesimpulan

Berdasarkan hasil yang telah diperoleh, maka kesimpulan yang bisa ditarik dari analisis dan pembahasan sebelumnya adalah dengan menggunakan 500 data, model Wikipedia dimensi vektor 300 dengan penambahan model pada dataset yang sudah dilakukan proses *case folding* dan juga *tokenizing*, dan melakukan perhitungan algoritma backpropagation dengan 3 hidden layer (128, 32, 8) dengan fungsi aktivasi ReLU didapatkan nilai MAPE terbaik yang didapatkan pada hasil pengujian proses training dan testing masing-masing 5% dan 8%. Dari hasil tersebut dapat dilihat bahwa model yang dihasilkan menggunakan algoritma backpropagation untuk melakukan pengkoreksian jawaban secara otomatis sangat baik. Selain itu pada pengujian kombinasi jumlah data training dan data testing terbaik terdapat pada kombinasi jumlah data training sebesar 75% atau 375 data dan jumlah data testing sebesar 25% atau 125 data.

B. Saran

Dari hasil yang di dapat maka peneliti memberi saran untuk penelitian selanjutnya dapat dilakukan pengujian lain terhadap arsitektur, jumlah iterasi, inisiasi bobot pada algoritma backpropagation atau melakukan optimasi algoritma backpropagation dengan menggunakan algoritma genetika, evolution strategy atau yang lainnya. Selain itu peneliti lain juga bisa menggunakan metode lain untuk membandingkan hasil pada penelitian ini pada kasus pengkoreksian jawaban essay otomatis.

REFENSI

- [1] H. Budiman, "Peran Teknologi Informasi Dan Komunikasi Dalam Pendidikan," *Al-Tadzkiyyah J. Pendidik. Islam*, vol. 8, no. 1, p. 31, 2017, doi: 10.24042/atjpi.v8i1.2095.
- [2] S. S. W. Lubis, "Keterampilan Menulis Essai Dalam Pembentukan Berpikir Kritis Mahasiswa Prodi PGMI UIN Ar-Raniry Banda Aceh," *PIONIR J. Pendidik.*, vol. 8, no. 2, pp. 1–17, 2019, [Online]. Available: <https://jurnal.ar-raniry.ac.id/index.php/Pionir/index>.
- [3] J. Bhattacharjee, *fastText Quick Start Guide: Get started with Facebook's library for text representation and classification*. Packt Publishing, 2018.

- [4] D. Shen *et al.*, "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms," *ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap.)*, vol. 1, pp. 440–450, 2018, doi: 10.18653/v1/p18-1041.
- [5] B. Athiwaratkun, A. G. Wilson, and A. Anandkumar, "Probabilistic fasttext for multi-sense word embeddings," *ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap.)*, vol. 1, pp. 1–11, 2018, doi: 10.18653/v1/p18-1001.
- [6] Y. D. Prabowo, T. L. Marselino, and M. Suryawiguna, "Pembentukan Vector Space Model Bahasa Indonesia Menggunakan Metode Word to Vector," *J. Buana Inform.*, vol. 10, no. 1, p. 29, 2019, doi: 10.24002/jbi.v10i1.2053.
- [7] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Trans. Assoc. Comput. Linguist.*, vol. 5, pp. 135–146, 2017, doi: 10.1162/tacl_a_00051.
- [8] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "FastText.zip: Compressing text classification models," pp. 1–13, 2016, [Online]. Available: <http://arxiv.org/abs/1612.03651>.
- [9] A. Raffi, M. Mulki, M. A. Bijaksana, A. A. Suryani, and T. Batasannya, "Analisis Model Word2vec dalam Penyelesaian Soal Analogi pada Bahasa Indonesia Pendahuluan Studi Terkait," vol. 6, no. 2, pp. 8513–8519, 2019.
- [10] M. Z. Sarwani and D. A. Sani, "Social Media Analysis Using Probabilistic Neural Network Algorithm to Know Personality Traits," *Inf. J. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 6, no. 1, pp. 61–64, 2021, doi: 10.25139/inform.v6i1.3307.
- [11] Munir, L. S. Riza, and A. Mulyadi, "An Automatic Scoring System for Essay by Using Methods Combination of Term Frequency and n-Gram," *Int. J. Trend Res. Dev.*, vol. 3, no. 6, pp. 403–407, 2016, [Online]. Available: www.ijtrd.com.
- [12] A. A. Putri Ratna, H. Khairunissa, A. Kaltsum, I. Ibrahim, and P. D. Purnamasari, "Automatic Essay Grading for Bahasa Indonesia with Support Vector Machine and Latent Semantic Analysis," *ICECOS 2019 - 3rd Int. Conf. Electr. Eng. Comput. Sci. Proceeding*, no. 2, pp. 363–367, 2019, doi: 10.1109/ICECOS47637.2019.8984528.
- [13] A. Amalia, D. Gunawan, Y. Fithri, and I. Aulia, "Automated Bahasa Indonesia essay evaluation with latent semantic analysis," *J. Phys. Conf. Ser.*, vol. 1235, no. 1, 2019, doi: 10.1088/1742-6596/1235/1/012100.
- [14] R. A. Rajagede, "Improving Automatic Essay Scoring for Indonesian Language using Simpler Model and Richer Feature," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, pp. 11–18, 2021, doi: 10.22219/kinetik.v6i1.1196.
- [15] D. Hindarto and H. Santoso, "NON-NEURAL NETWORK AND NEURAL NETWORK Jurnal Nasional Pendidikan Teknik Informatika : JANAPATI | 50," vol. 11, pp. 49–62, 2022.